

new
March 29, 2001
Tomohiko Ogishima
0965-0343P
BSKB
703-205-8000

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

11033 U.S. PTO
09/820008
03/29/01

出 願 年 月 日
Date of Application: 2000年 3月29日

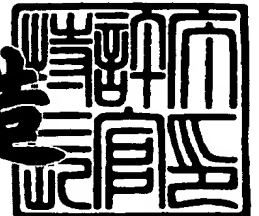
出 願 番 号
Application Number: 特願2000-090661

出 願 人
Applicant(s): ケイディディ株式会社

2000年10月13日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3084023

【書類名】 特許願

【整理番号】 K00020901

【特記事項】 特許法第 3 0 条第 1 項の規定の適用を受けようとする特
許出願

【提出日】 平成12年 3月29日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 12/28
H04Q 3/00

【発明者】

【住所又は居所】 埼玉県上福岡市大原二丁目 1 番 1 5 号 株式会社ケイデ
ィディ研究所内

【氏名】 大岸 智彦

【発明者】

【住所又は居所】 埼玉県上福岡市大原二丁目 1 番 1 5 号 株式会社ケイデ
ィディ研究所内

【氏名】 井戸上 彰

【発明者】

【住所又は居所】 埼玉県上福岡市大原二丁目 1 番 1 5 号 株式会社ケイデ
ィディ研究所内

【氏名】 長谷川 亨

【発明者】

【住所又は居所】 埼玉県上福岡市大原二丁目 1 番 1 5 号 株式会社ケイデ
ィディ研究所内

【氏名】 加藤 聰彦

【特許出願人】

【識別番号】 000001214

【氏名又は名称】 ケイディディ株式会社

【代理人】

【識別番号】 100078499

【弁理士】

【氏名又は名称】 光石 俊郎

【電話番号】 03-3583-7058

【選任した代理人】

【識別番号】 100074480

【弁理士】

【氏名又は名称】 光石 忠敬

【電話番号】 03-3583-7058

【選任した代理人】

【識別番号】 100102945

【弁理士】

【氏名又は名称】 田中 康幸

【電話番号】 03-3583-7058

【手数料の表示】

【予納台帳番号】 020318

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 トラヒック統計情報収集方法

【特許請求の範囲】

【請求項 1】 インターネット回線上の一方向のトラヒックから、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するDATAセグメントを検出すること、及び、
発側転送セグメント数として、検出したDATAセグメントの総数を計算し、あるいは、発側転送バイト数として、最初に検出したDATAセグメントのシーケンス番号と、最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算し、あるいは、前記発側転送セグメント数及び前記発側転送バイト数の両方を検出すること
を特徴とするトラヒック統計情報収集方法。

【請求項 2】 インターネット回線上の一方向のトラヒックから、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するDATAセグメントを検出すること、及び、
着側転送セグメント数として、検出したDATAセグメントの総数を計算し、あるいは、着側転送バイト数として、最初に検出したDATAセグメントのシーケンス番号と、最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算し、あるいは、前記着側転送セグメント数及び前記着側転送バイト数の両方を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項 3】 インターネット回線上の一方向のトラヒックから、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出すること、及び、
発側データ受信量として、最初に検出したACKセグメント又はDATAセグメントの応答確認番号と最後に検出したACKセグメント又はDATAセグメントの応答確認番号との差を計算すること
を特徴とするトラヒック統計情報収集方法。

【請求項 4】 インターネット回線上の一方向のトラヒックから、SYN+

ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出すること、及び、

着側データ受信量として、最初に検出したACKセグメント又はDATAセグメントの応答確認番号と最後に検出したACKセグメント又はDATAセグメントの応答確認番号との差を計算すること
を特徴とするトラフィック統計情報収集方法。

【請求項5】 インターネット回線上の一方向のトラフィックから、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するDATAセグメントを検出すること、

DATAセグメントの検出毎に、検出したDATAセグメントのシーケンス番号から、次に送られてくるべきDATAセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求めること、

DATAセグメントの検出毎に、新たに検出したDATAセグメントのシーケンス番号が前回のDATAセグメントの検出で求めた次シーケンス番号より小さいか否か判定すること、及び、

前記判定が肯定である場合に、今までの発側再送セグメント数に1を加算して新たな発側再送セグメント数を求め、あるいは、今までの発側再送バイト数に、前記次シーケンス番号と前記新たに検出したDATAセグメントのシーケンス番号との差と、前記新たに検出したDATAセグメントのユーザデータ長とのうち、小さい方を加算して新たな発側再送バイト数を求め、あるいは、前記新たな発側再送セグメント数及び前記新たな発側再送バイト数の両方を求めること
を特徴とするトラフィック統計情報収集方法。

【請求項6】 インターネット回線上の一方向のトラフィックから、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するDATAセグメントを検出すること、

DATAセグメントの検出毎に、検出したDATAセグメントのシーケンス番号から、次に送られてくるべきDATAセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求めること、

D A T A セグメントの検出毎に、新たに検出した D A T A セグメントのシーケンス番号が前回の D A T A セグメントの検出で求めた次シーケンス番号より小さいか否かを判定すること、及び、

前記判定が肯定である場合に、今までの着側再送セグメント数に 1 を加算して新たな着側再送セグメント数を求め、あるいは、今までの着側再送バイト数に、前記次シーケンス番号と前記新たに検出した D A T A セグメントのシーケンス番号との差と、前記新たに検出した D A T A セグメントのユーザデータ長とのうち、小さい方を加算して新たな着側再送バイト数を求め、あるいは、前記新たな着側再送セグメント数及び前記新たな着側再送バイト数の両方を求めることを特徴とするトラフィック統計情報収集方法。

【請求項 7】 インターネット回線上の一方向のトラフィックから、S Y N セグメントを検出し、更に、検出した S Y N セグメントと同じコネクションに属する A C K セグメントを検出すること、

A C K セグメントの検出毎に、検出した A C K セグメントの応答確認番号及びウィンドウサイズから、次に送られてくるべき A C K セグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つ A C K セグメントのウィンドウサイズ（以下、現ウィンドウサイズと呼ぶ）を求めること、

A C K セグメントの検出毎に、新たに検出した A C K セグメントの応答確認番号及びウィンドウサイズの両方が前回の A C K セグメントの検出で求めた次応答確認番号及び現ウィンドウサイズに等しいか否かを判定すること、及び、

応答確認番号及びウィンドウサイズの組が同じ値の 2 つ以上の A C K セグメントについて前記判定が肯定である毎に、今までの発側 D A T A セグメント紛失数に 1 を加算して新たな発側 D A T A セグメント紛失数を求めること

を特徴とするトラフィック統計情報収集方法。

【請求項 8】 インターネット回線上の一方向のトラフィックから、S Y N + A C K セグメントを検出し、更に、検出した S Y N + A C K セグメントと同じコネクションに属する A C K セグメントを検出すること、

A C K セグメントの検出毎に、検出した A C K セグメントの応答確認番号及びウィンドウサイズから、次に送られてくるべき A C K セグメントの応答確認番号（

以下、次応答確認番号と呼ぶ) と、最大の応答確認番号を持つ A C K セグメントのウィンドウサイズ (以下、現ウィンドウサイズと呼ぶ) を求めること、
 A C K セグメントの検出毎に、新たに検出した A C K セグメントの応答確認番号及びウィンドウサイズの両方が前回の A C K セグメントの検出で求めた次応答確認番号及び前記現ウィンドウサイズに等しいか否かを判定すること、及び、
 応答確認番号及びウィンドウサイズの組が同じ値の 2 つ以上の A C K セグメントについて前記判定が肯定である毎に、今までの着側 D A T A セグメント紛失数に 1 を加算して新たな着側 D A T A セグメント紛失数を求めること
 を特徴とするトラフィック統計情報収集方法。

【請求項 9】 インターネット回線上の一方向のトラフィックから、S Y N セグメントを検出し、更に、検出した S Y N セグメントと同じコネクションに属する連続複数の D A T A セグメント及び同 D A T A セグメントに続く連続複数の A C K セグメントを検出すること、及び、
 発側 H T T P 応答時間として、最後の D A T A セグメントの検出からの最初の A C K セグメント検出までの時刻差を計算し、あるいは、発側 H T T P スループットとして、最初に検出した A C K セグメントと最後に検出した A C K セグメントとの応答確認番号の差と、最初の A C K セグメントの検出から最後の A C K セグメントの検出までの時刻差の比を計算し、あるいは、前記発側 H T T P 応答時間及び前記発側 H T T P スループットの両方を計算すること
 を特徴とするトラフィック統計情報収集方法。

【請求項 1 0】 インターネット回線上の一方向のトラフィックから、S Y N + A C K セグメントを検出し、更に、検出した S Y N + A C K セグメントと同じコネクションに属する連続複数の A C K セグメント及び同 A C K セグメントに続く連続複数の D A T A セグメントを検出すること、
 着側 H T T P 応答時間として、最後の A C K セグメントの検出から最初の D A T A セグメントの検出までの時刻差を計算し、あるいは、着側 H T T P スループットとして、最初に検出した D A T A セグメントのシーケンス番号と最後に検出した D A T A セグメントのシーケンス番号にそのユーザデータ長を加算した値との差と、最初の D A T A セグメントの検出から最後の D A T A セグメントの検出

までの時刻差との比を計算し、あるいは、前記着側 H T T P 応答時間及び前記着側 H T T P スループットの両方を計算すること
を特徴とするトラフィック統計情報収集方法。

【請求項 1 1】 インターネット回線上の一方向のトラフィックから、SYN + ACK セグメントを検出し、更に、検出した SYN + ACK セグメントと同じコネクションに属する連続複数の ACK セグメントを検出すること、及び、
発側 F T P スループットとして、最初に検出した ACK セグメントと最後に検出した ACK セグメントとの応答確認番号の差と、最初の ACK セグメントの検出から最後の ACK セグメントの検出までの時刻差との比を計算すること
を特徴とするトラフィック統計情報収集方法。

【請求項 1 2】 インターネット回線上の一方向のトラフィックから、SYN セグメントを検出し、更に、検出した SYN セグメントと同じコネクションに属する連続複数の DATA セグメントを検出すること、及び、
着側 F T P スループットとして、最初に検出した DATA セグメントのシーケンス番号と最後に検出した DATA セグメントのシーケンス番号にそのユーザデータ長を加算した値との差と、最初の DATA セグメントの検出から最後の DATA セグメントの検出までの時刻差との比を計算すること
を特徴とするトラフィック統計情報収集方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明はインターネット回線上を流れる T C P / I P のトラフィックをキャプチャ（取得）し、それを基にトラフィックの統計情報を収集する技術に関し、特に、片方向のトラフィックのみ取得できる場合において、T C P レベルの情報を管理することにより、トラフィックが取得できなかった方向のトラフィック統計情報の収集を可能にするものである。

【0 0 0 2】

【従来の技術】

インターネットの普及に伴い、データ転送量等、トラフィックの統計情報を測定

するパフォーマンス解析が重要となっている。これに応え、特開平 1 1 - 2 5 2 1 1 1 号公報に開示されたように、本発明者らは、TCP レベルのトラフィック統計情報（再送データ量やコネクション確立の待ち時間など）を測定することによりインターネットの状況を判別できると考え、バックボーン回線等をモニタし IP レベルのトラフィック統計情報に加えて、TCP レベルも併せて収集することが可能なトラフィック監視装置を既に開発した。このトラフィック監視装置は、TCP 通信の双方向のセグメントが同一回線でモニタ可能であることを前提する。

【0003】

しかしながら、IP ルーティングは方向別に制御されるため、必ずしも双方向のトラフィックが同一回線を経由するとは限らない。つまり、現実のネットワーク構成においては、行き帰りのルートが違う場合が考えられるから、必ずしも双方向のトラフィックを取得できるとは限らない。

【0004】

その他、ルータにおいて単位時間毎のセグメント数やバイト数などの IP レベルのトラフィック統計情報を収集する技術（MR TG と呼ばれる）が知られている。また、IP レベルのトラフィック統計情報に加え、アプリケーション別のセグメント数やバイト数などの TCP レベルのトラフィック統計情報を収集する技術（S N I F F E R と呼ばれる）も知られている。しかし、いずれも TCP 通信の双方向のセグメントが同一回線でモニタ可能であることを前提とする。

【0005】

【発明が解決しようとする課題】

本発明の目的は、片方向のトラフィックしか取得できない場合においても、そのトラフィックを用いて TCP レベルの情報を管理することにより、双方向、特に、トラフィックが取得できなかった方向のトラフィック統計情報を収集することを可能にすることである。

【0006】

【課題を解決するための手段】

請求項 1 に係る発明は、インターネット回線上の一方向のトラフィックからセグメントを取得し、発側データ転送量を収集するトラフィック統計情報収集方法であ

り、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するDATAセグメントを検出し、発側転送セグメント数として、検出したDATAセグメントの総数を計算し、あるいは、発側転送バイト数として、最初に検出したDATAセグメントのシーケンス番号と、最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算し、あるいは、前記発側転送セグメント数及び前記発側転送バイト数の両方を検出する。

請求項2に係る発明は、着側データ転送量を収集するトラフィック統計情報収集方法であり、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するDATAセグメントを検出し、着側転送セグメント数として、検出したDATAセグメントの総数を計算し、あるいは、着側転送バイト数として、最初に検出したDATAセグメントのシーケンス番号と、最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算し、あるいは、前記着側転送セグメント数及び前記着側転送バイト数の両方を計算する。

【0007】

請求項3に係る発明は、インターネット回線上の一方向のトラフィックからセグメントを取得し、発側データ受信量を収集するトラフィック統計情報収集方法であり、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出し、発側データ受信量として、最初に検出したACKセグメント又はDATAセグメントの応答確認番号と最後に検出したACKセグメント又はDATAセグメントの応答確認番号との差を計算する。

請求項4に係る発明は、着側データ受信量を収集するトラフィック統計情報収集方法であり、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出し、着側データ受信量として、最初に検出したACKセグメント又はDATAセグメントの応答確認番号と最後に検出したACKセグメント又はDATAセグメントの応答確認番号との差を計算する。

【 0 0 0 8 】

請求項 5 に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側再送量を収集するトラヒック統計情報収集方法であり、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するDATAセグメントを検出し、DATAセグメントの検出毎に、検出したDATAセグメントのシーケンス番号から、次に送られてくるべきDATAセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求め、DATAセグメントの検出毎に、新たに検出したDATAセグメントのシーケンス番号が前回のDATAセグメントの検出で求めた次シーケンス番号より小さいか否か判定し、前記判定が肯定である場合に、今までの発側再送セグメント数に1を加算して新たな発側再送セグメント数を求め、あるいは、今までの発側再送バイト数に、前記次シーケンス番号と前記新たに検出したDATAセグメントのシーケンス番号との差と、前記新たに検出したDATAセグメントのユーザデータ長とのうち、小さい方を加算して新たな発側再送バイト数を求め、あるいは、前記新たな発側再送セグメント数及び前記新たな発側再送バイト数の両方を求める。

請求項 6 に係る発明は、着側再送量を収集するトラヒック統計情報収集方法であり、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するDATAセグメントを検出し、DATAセグメントの検出毎に、検出したDATAセグメントのシーケンス番号から、次に送られてくるべきDATAセグメントのシーケンス番号（以下、次シーケンス番号と呼ぶ）を求め、DATAセグメントの検出毎に、新たに検出したDATAセグメントのシーケンス番号が前回のDATAセグメントの検出で求めた次シーケンス番号より小さいか否か判定し、前記判定が肯定である場合に、今までの着側再送セグメント数に1を加算して新たな着側再送セグメント数を求め、あるいは、今までの着側再送バイト数に、前記次シーケンス番号と前記新たに検出したDATAセグメントのシーケンス番号との差と、前記新たに検出したDATAセグメントのユーザデータ長とのうち、小さい方を加算して新たな着側再送バイト数を求め、あるいは、前記新たな着側再送セグメント数及び前記新たな着側再送バイト数の両方を求める。

【 0 0 0 9 】

請求項 7 に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側 DATA セグメント紛失数を収集するトラヒック統計情報収集方法であり、SYN セグメントを検出し、更に、検出した SYN セグメントと同じコネクションに属する ACK セグメントを検出し、ACK セグメントの検出毎に、検出した ACK セグメントの応答確認番号及びウィンドウサイズから、次に送られてくるべき ACK セグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つ ACK セグメントのウィンドウサイズ（以下、現ウィンドウサイズと呼ぶ）を求め、ACK セグメントの検出毎に、新たに検出した ACK セグメントの応答確認番号及びウィンドウサイズの両方が前回の ACK セグメントの検出で求めた次応答確認番号及び現ウィンドウサイズに等しいか否かを判定し、応答確認番号及びウィンドウサイズの組が同じ値の 2 つ以上の ACK セグメントについて前記判定が肯定である毎に、今までの発側 DATA セグメント紛失数に 1 を加算して新たな発側 DATA セグメント紛失数を求める。

請求項 8 に係る発明は、着側 DATA セグメント紛失数を収集するトラヒック統計情報収集方法であり、SYN+ACK セグメントを検出し、更に、検出した SYN+ACK セグメントと同じコネクションに属する ACK セグメントを検出し、ACK セグメントの検出毎に、検出した ACK セグメントの応答確認番号及びウィンドウサイズから、次に送られてくるべき ACK セグメントの応答確認番号（以下、次応答確認番号と呼ぶ）と、最大の応答確認番号を持つ ACK セグメントのウィンドウサイズ（以下、現ウィンドウサイズと呼ぶ）を求め、ACK セグメントの検出毎に、新たに検出した ACK セグメントの応答確認番号及びウィンドウサイズの両方が前回の ACK セグメントの検出で求めた次応答確認番号及び前記現ウィンドウサイズに等しいか否かを判定し、応答確認番号及びウィンドウサイズの組が同じ値の 2 つ以上の ACK セグメントについて前記判定が肯定である毎に、今までの着側 DATA セグメント紛失数に 1 を加算して新たな着側 DATA セグメント紛失数を求める。

【 0 0 1 0 】

請求項 9 に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側の HTTP レベルの情報を収集するトラヒック統計情報収集方法であり、SYN セグメントを検出し、更に、検出した SYN セグメントと同じコネクションに属する連続複数の DATA セグメント及び同 DATA セグメントに続く連続複数の ACK セグメントを検出し、発側 HTTP 応答時間として、最後の DATA セグメントの検出から最初の ACK セグメントの検出までの時刻差を計算し、あるいは、発側 HTTP スループットとして、最初に検出した ACK セグメントと最後に検出した ACK セグメントとの応答確認番号の差と、最初の ACK セグメントの検出から最後の ACK セグメントの検出までの時刻差の比を計算し、あるいは、前記発側 HTTP 応答時間及び前記発側 HTTP スループットの両方を計算する。

請求項 10 に係る発明は、着側の HTTP レベルの情報を収集するトラヒック統計情報収集方法であり、SYN+ACK セグメントを検出し、更に、検出した SYN+ACK セグメントと同じコネクションに属する連続複数の ACK セグメント及び同 ACK セグメントに続く連続複数の DATA セグメントを検出し、着側 HTTP 応答時間として、最後の ACK セグメントの検出から最初の DATA セグメントの検出までの時刻差を計算し、あるいは、着側 HTTP スループットとして、最初に検出した DATA セグメントのシーケンス番号と最後に検出した DATA セグメントのシーケンス番号にそのユーザデータ長を加算した値との差と、最初の DATA セグメントの検出から最後の DATA セグメントの検出までの時刻差との比を計算し、あるいは、前記着側 HTTP 応答時間及び前記着側 HTTP スループットの両方を計算する。

上記いずれの場合も、前記 ACK セグメントを連続して所定数以上、例えば 10 以上検出した場合のみ、前記発側 HTTP スループットあるいは前記着側 HTTP スループットの計算を行うことにより、異常値を効果的に排除することができる。

【 0 0 1 1 】

請求項 11 に係る発明は、インターネット回線上の一方向のトラヒックからセグメントを取得し、発側の FTP レベルの情報を収集するトラヒック統計情報収

集方法であり、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属する連続複数のACKセグメントを検出し、発側FTPスループットとして、最初に検出したACKセグメントと最後に検出したACKセグメントとの応答確認番号の差と、最初のACKセグメントの検出から最後のACKセグメントの検出までの時刻差との比を計算する。この場合、前記ACKセグメントを連続して所定数以上、例えば10以上検出した場合のみ、前記発側FTPスループットの計算を行うことにより、異常値を効果的に排除することができる。

請求項12に係る発明は、着側のFTPレベルの情報を収集するトラフィック統計情報収集方法であり、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属する連続複数のDATAセグメントを検出し、着側FTPスループットとして、最初に検出したDATAセグメントのシーケンス番号と最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差と、最初のDATAセグメントの検出から最後のDATAセグメントの検出までの時刻差との比を計算する。この場合、前記DATAセグメントを連続して所定数以上、例えば10以上検出した場合のみ、前記着側FTPスループットの計算を行うことにより、異常値を効果的に排除することができる。

【0012】

【発明の実施の形態】

以下、図面を参照して、本発明の実施形態例を説明する。

【0013】

図1に、本発明を実現する双方向トラフィック統計情報収集装置の構成例を示す。図1に示す装置1は、セグメント取得モジュール2と、解析モジュール3と、蓄積モジュール4と、状態遷移表の格納部5と、統計情報の格納部6を有している。この装置1が収集するトラフィック統計情報は、大別するとTCPレベルの情報及びアプリケーションレベル（HTTPレベル及びFTPレベル）の情報であり、以下に詳細を示す。ここで、発側とはSYNセグメントの送信側、着側とはSYN+ACKセグメントの送信側を意味する。但し、FTPのデータ用TCP

コネクションではサーバがSYNセグメントを送信し、クライアントがSYN+ACKセグメントを送信するので、FTPでの発側とはSYN+ACKセグメントの送信側、着側とはSYNセグメントの送信側を意味する。

(1) TCPレベルの情報には、下記のものがある。

発側データ転送量 (セグメント数/バイト数)
 着側データ転送量 (セグメント数/バイト数)
 発側データ受信量 (バイト数)
 着側データ受信量 (バイト数)
 発側再送量 (セグメント数/バイト数)
 着側再送量 (セグメント数/バイト数)
 発側DATAセグメント紛失数 (セグメント数)
 着側DATAセグメント紛失数 (セグメント数)

(2) HTTPレベルの情報には、下記のものがある。

発側応答時間
 発側スループット
 着側応答時間
 着側スループット

(3) FTPレベルの情報には、下記のものがある。

発側スループット
 着側スループット

【0014】

セグメント取得モジュール2は、インターネット回線（例えばバックボーン等、インターネットの一部をなす回線）15に接続され、同回線上のトラフィックを例えば一定時間毎にモニタしてセグメント（パケットと呼ばれることもある）を取得し、解析モジュール3に送る。

【0015】

図1に示すネットワークでは、A端末7はルータ11とこのルータ11に接続された2つのルータ12、13を介してインターネット10に接続され、B端末8と端末9は共通のルータ14を介してインターネット10に接続され、セグメ

ント取得モジュール2はルータ11、12間でインターネット回線15に接続されている。つまり、A端末7とインターネット10の接続ルートは2通り、ルータ11とルータ12を介したものと、ルータ11とルータ13を介してものがある。

【0016】

今、図1のネットワークにおいて、A端末7からB端末8へのトラフィックが実線16aで示すようにルータ11からルータ12、インターネット10、ルータ14を順に通るものであり、B端末8からA端末7へのトラフィックは実線16bで示すようにルータ14からインターネット10、ルータ13、ルータ11を順に通るものである場合を考える。この場合、セグメント取得モジュール2は、A端末7とB端末8間のトラフィックのうち、実線16aで示す一方向のトラフィック（A端末7からB端末8へのトラフィック）のみしか取得できない。

また、C端末9からA端末7へのトラフィックが破線17aで示すようにルータ14からインターネット10、ルータ12、ルータ11を順に通るものであり、A端末7からC端末9からへのトラフィックは破線17bで示すようにルータ11からルータ13、インターネット10、ルータ14を順に通るものである場合を考える。この場合は、セグメント取得モジュール2は、C端末9とA端末7間のトラフィックのうち、破線17aで示す一方向のトラフィック（C端末9からA端末7へのトラフィック）のみしか取得できない。

【0017】

解析モジュール3はセグメント取得モジュール3が取得したセグメントから、データ転送量等、トラフィック統計情報を計算する。

【0018】

その際、解析モジュール3は、TCPヘッダ中のフラグビット6種類から、どの種類のセグメントを検出したかを識別する。

【0019】

ここで、SYNビットのみ1となっているセグメントをSYNセグメントと呼び、SYNビットとACKビットのみが共に1となっているセグメントをSYN+ACKセグメントと呼ぶ。

【 0 0 2 0 】

また、SYNビット、FINビット及びACKビットの3種類において、ACKビットのみが1となっているセグメントのうち、ユーザデータが存在するものをDATAセグメントと呼び、ユーザデータが存在しないものをACKセグメントと呼ぶ。

【 0 0 2 1 】

更に、解析モジュール3は、取得したセグメントがどのコネクションに属するかを、例えば、取得したセグメントの発側IPアドレス及び着側IPアドレスに基づいて識別する。

【 0 0 2 2 】

また、解析モジュール3は、取得したセグメントがDATAセグメントであればそのシーケンス番号、ユーザデータ長及び応答確認番号を認識し、取得したセグメントがACKセグメントであればその応答確認番号を認識する。

【 0 0 2 3 】

更に、解析モジュール3は、格納部5の状態遷移表を参照して、DATAセグメントを検出する毎に、今までに検出したDATAセグメントのシーケンス番号から、次に送られてくるべきDATAセグメントのシーケンス番号、つまり次シーケンス番号seq__nxtを求めて管理する。この次シーケンス番号seq__nxtは、今までに転送された最大のシーケンス番号と同じである。また、ACKセグメントを検出する毎に、今までに検出したACKセグメントの応答確認番号から、次に送られてくるべきACKセグメントの応答確認番号、つまり次応答確認番号ack__nxtを求めて管理し、また、最大の応答確認番号が検出された時のACKセグメントが持つウィンドウサイズを、現ウィンドウサイズwinとして求めて管理する。

【 0 0 2 4 】

蓄積モジュール4は、解析モジュール3の計算で得たトラフィック統計情報を、IPアドレス別や、コネクション別、アプリケーション別に格納部6に格納する。

【 0 0 2 5 】

以下に、トラフィック統計情報収集の基礎となる解析モジュール3における計算例を説明する。ここで、取得したトラフィックがSYNセグメント送信側のトラフィック（発側トラフィック）か、SYN+ACKセグメント送信側のトラフィック（着側トラフィック）かを区別することにより、クライアントからの通信か、サーバからの通信かを識別することができる。

【0026】

[発側データ転送量の収集]

図2に示すシーケンスを参照して、発側データ転送量の収集を説明する。発側データ転送量としては発側転送セグメント数と発側転送バイト数があり、これらの収集には、モニタ可能な方向のトラフィックで、SYNセグメントを検出し、更に、このSYNセグメントと同じコネクションに属するDATAセグメントを検出する。

【0027】

図2に示すように、モニタ可能な方向でSYNセグメントが検出された場合、連続して検出されたDATAセグメントの数を計算することにより、発側転送セグメント数を求める。

【0028】

また、図2に示すように、SYNセグメントが検出された場合、連続して検出されたDATAセグメントのうち、最初のDATAセグメントのシーケンス番号SEQ1と、最後のDATAセグメントのシーケンス番号SEQnと、最後のDATAセグメントのユーザデータ長LENnとを用い、式(1)を計算することにより、発側転送バイト数ini_sdtを求める。

$$ini_sdt = (SEQn + LENn) - SEQ1 \quad \cdots \text{式(1)}$$

即ち、発側転送バイト数ini_sdtは、最初に検出したDATAセグメントのシーケンス番号SEQ1と、最後に検出したDATAセグメントのシーケンス番号SEQnにそのユーザデータ長LENnを加算した値(SEQn+LENn)との差である。

【0029】

発側転送セグメント数と発側転送バイト数のうち、一方のみを発側データ転送

量として計算するようにしても良い。

【0030】

[着側データ転送量の収集]

図3に示すシーケンスを参照して、着側データ転送量の収集を説明する。着側データ転送量としては着側転送セグメント数と着側転送バイト数があり、これらの収集には、モニタ可能な方向のトラヒックで、SYN+ACKセグメントを検出し、更に、このSYN+ACKセグメントと同じコネクションに属するDATAセグメントを検出する。

【0031】

図3に示すように、モニタ可能な方向でSYN+ACKセグメントが検出された場合、連続して検出されたDATAセグメントを計算することにより、着側転送セグメント数を求める。

【0032】

また、図3に示すように、SYN+ACKセグメントが検出された場合、連続して検出されたDATAセグメントのうち、最初のDATAセグメントのシーケンス番号SEQ1と、最後のDATAセグメントのシーケンス番号SEQnと、そのユーザデータ長LENnとを用い、式(2)を計算して、着側転送バイト数rsp_sdtを求める。

$$rsp_sdt = (SEQn + LENn) - SEQ1 \quad \cdots \text{式(2)}$$

即ち、着側転送バイト数rsp_sdtは、最初に検出したDATAセグメントのシーケンス番号SEQ1と、最後に検出したDATAセグメントのシーケンス番号SEQnにそのユーザデータ長LENnを加算した値(SEQn+LENn)との差である。

【0033】

この場合も、着側転送セグメント数と着側転送バイト数のうち、一方のみを着側データ転送量として計算するようにしても良い。

【0034】

[発側データ受信量の収集]

図4に示すシーケンスを参照して、発側データ受信量(バイト数)の収集を説

明する。発側データ受信量の収集には、モニタ可能な方向のトラヒックで、SYNセグメントを検出し、更に、このSYNセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出する。

【0035】

図4に示すように、モニタ可能な方向でSYNセグメントが検出された場合、連続して検出されたACKセグメントあるいはDATAセグメントのうち、最初のACKセグメントあるいはDATAセグメントの応答確認番号ACK1と、最後のACKセグメントあるいはDATAセグメントの応答確認番号ACKnとを用い、式(3)を計算することにより、発側データ受信量 ini_rdt とする。

$$ini_rdt = ACKn - ACK1 \quad \dots \text{式(3)}$$

即ち、発側受信バイト数 ini_rdt は、最初に検出したACKセグメントあるいはDATAセグメントの応答確認番号ACK1と、最後に検出したACKセグメントあるいはDATAセグメントの応答確認番号ACKnとの差である。

【0036】

〔着側データ受信量の収集〕

図5に示すシーケンスを参照して、着側データ受信量(バイト数)の収集を説明する。着側データ受信量の収集には、モニタ可能な方向のトラヒックで、SYN+ACKセグメントを検出し、更に、このSYN+ACKセグメントと同じコネクションに属するACKセグメント及びDATAセグメントを検出する。

【0037】

図5に示すように、モニタ可能な方向でSYN+ACKセグメントが検出された場合、連続して検出されたACKセグメントあるいはDATAセグメントのうち、最初のACKセグメントあるいはDATAセグメントの応答確認番号ACK1と、最後のACKセグメントあるいはDATAセグメントの応答確認番号ACKnとを用い、式(4)を計算することにより、着側データ受信量 rsp_rdt とする。

$$rsp_rdt = ACKn - ACK1 \quad \dots \text{式(4)}$$

即ち、着側受信バイト数 rsp_rdt は、最初に検出したACKセグメントあ

るいは DATA セグメントの応答確認番号 ACK 1 と、最後に検出した ACK セグメントあるいは DATA セグメントの応答確認番号 ACK n との差である。

【0038】

次に、図 6 を参照して、発側再送量及び着側再送量の収集を説明する。発側再送量及び着側再送量にはそれぞれ再送セグメント数と再送バイト数がある。

【0039】

[発側再送量の収集]

発側再送量の収集には、モニタ可能な方向のトラヒックで、先ず、SYN セグメントを検出する（図示省略）。更に、この SYN セグメントと同じコネクションに属する DATA セグメントを検出する（図 6（a）参照）。

【0040】

モニタ可能な方向で、SYN セグメントに続いて DATA セグメントが新たに検出された場合、この新たに検出した DATA セグメントのシーケンス番号 SEQ が前述の別途管理により前回までの DATA セグメントの検出で求めた次シーケンス番号 seq__next より小さいか否か判定し、この判定が肯定（SEQ < seq__next）である場合に、SYN セグメントの送信側から再送があったと判断する。

【0041】

発側再送量のうち、発側再送セグメント数 ini__ret__num は、式（5）に示すように、今までの発側再送セグメント数に 1 を加算して求める。

$$\text{ini_ret_num} = \text{ini_ret_num} + 1 \quad \cdots \text{式（5）}$$

【0042】

発側再送バイト数 ini__ret は、式（6）に示すように、重複のあった分、例えば図 6（b）にて斜線を付した部分 18、19 のバイト数を今までの発側再送バイト数に加算して求める。

$$\begin{aligned} \text{ini_ret} &= \text{ini_ret} \\ &+ \min(\text{seq_next} - \text{SEQ}, \text{LEN}) \quad \cdots \text{式（6）} \end{aligned}$$

即ち、発側再送バイト数 ini__ret は、次シーケンス番号 seq__next と新たに検出した DATA セグメントのシーケンス番号 SEQ との差（seq__n

x t - S E Q) と、この新たに検出した DATA セグメントのユーザデータ長 L E N とのうち、小さい方を今までの発側再送バイト数に加算した値である。

【 0 0 4 3 】

発側再送セグメント数と発側再送バイト数のうち、一方のみを発側再送量として計算するようにしても良い。

【 0 0 4 4 】

図 6 (a) に示す例では、2 番目の DATA セグメント (S E Q = 1 4 6 0 , L E N = 1 4 6 0) を検出した時点で、次シーケンス番号 s e q _ n x t は 2 9 2 0 であるため、3 番目の DATA セグメント (S E Q = 0 , L E N = 1 4 6 0) の検出により、再送があったと判断する。この時、s e q _ n x t - S E Q = 2 9 2 0 、 L E N = 1 4 6 0 であるから、図 6 (b) に示す斜線部分 (重複分) 1 8 のバイト数は 1 4 6 0 バイトであり、これが発側再送セグメント数とされる。

【 0 0 4 5 】

また、図 6 (a) に示す例では、4 番目の DATA セグメント (S E Q = 2 9 2 0 , L E N = 1 4 6 0) を検出した時点で、次シーケンス番号 s e q _ n x t は 4 3 8 0 であるため、5 番目の DATA セグメント (S E Q = 4 0 0 0 , L E N = 1 4 6 0) の検出により、再送があったと判断する。この時、s e q _ n x t - S E Q = 3 8 0 、 L E N = 1 4 6 0 であるから、図 6 (b) に示す斜線部分 (重複分) 1 9 のバイト数は 3 8 0 バイトであり、これを今までの発側再送セグメント数 1 4 6 0 に加算する。従って、新たな発側再送セグメント数は 1 8 4 0 となる。

【 0 0 4 6 】

発側再送セグメント数と発側再送バイト数のうち、一方のみを発側再送量として計算するようにしても良い。

【 0 0 4 7 】

[着側再送量の収集]

着側再送量の収集には、モニタ可能な方向のトラヒックで、先ず、S Y N + A C K セグメントを検出する (図示省略) 。更に、この S Y N + A C K セグメント

と同じコネクションに属するDATAセグメントを検出する（図6（a）参照）

【0048】

モニタ可能な方向で、SYN+ACKセグメントに続いてDATAセグメントが新たに検出された場合、この新たに検出したDATAセグメントのシーケンス番号SEQが前述の別途管理により前回までのDATAセグメントの検出で求めた次シーケンス番号seq_nextより小さいか否か判定し、この判定が肯定（ $SEQ < seq_next$ ）である場合に、SYN+ACKセグメントの送信側から再送があったと判断する。

【0049】

着側再送量のうち、着側再送セグメント数rsp_ret_numは、式（7）に示すように、今までの着側再送セグメント数に1を加算して求める。

$$rsp_ret_num = rsp_ret_num + 1 \quad \cdots \text{式（7）}$$

【0050】

着側再送バイト数rsp_retは、式（8）に示すように、重複のあった分、例えば図6（b）にて斜線を付した部分18、19のバイト数を今までの着側再送バイト数に加算して求める。

$$rsp_ret = rsp_ret + \min(seq_next - SEQ, LEN) \quad \cdots \text{式（8）}$$

即ち、着側再送バイト数rsp_retは、次シーケンス番号seq_nextと新たに検出したDATAセグメントのシーケンス番号SEQとの差（ $seq_next - SEQ$ ）と、この新たに検出したDATAセグメントのユーザデータ長LENとのうち、小さい方を今までの着側再送バイト数に加算した値である。

【0051】

この場合も、着側再送セグメント数と着側再送バイト数のうち、一方のみを着側再送量として計算するようにしても良い。

【0052】

次に、図7を参照して、発側DATAセグメント紛失数及び着側DATAセグメント紛失数の収集を説明する。

【0053】

[発側DATAセグメント紛失数の収集]

発側DATAセグメント紛失数の収集には、モニタ可能な方向のトラヒックで、先ず、SYNセグメントを検出する（図示省略）。更に、このSYNセグメントと同じコネクションに属するACKセグメントを検出する（図7参照）。

【0054】

モニタ可能な方向で、SYNセグメントに続いてACKセグメントが新たに検出された場合、この新たに検出したACKセグメントの応答確認番号ACK及びウィンドウサイズWINを、前述の別途管理している次応答確認番号ack__next及び現ウィンドウサイズwinとそれぞれ比較し、式（9）と式（10）の両方が成立するとき、DATAセグメント紛失があったと判断する。

$$ACK = ack_next \quad \cdots \text{式（9）}$$

$$WIN = win \quad \cdots \text{式（10）}$$

【0055】

但し、応答確認番号ACKとウィンドウサイズWINの組が同じ値のACKセグメントについて2回以上DATAセグメント紛失があったと判断した場合のみ、式（11）に示すように、今までの発側DATAセグメント紛失数に1を加算して、新たな発側DATAセグメント紛失数ini__recv__drop__numとする。

$$ini_recv_drop_num = ini_recv_drop_num + 1 \quad \cdots \text{式（11）}$$

【0056】

このような計算（応答確認番号ACK及びウィンドウサイズWINの組が同じ値の2つ以上のACKセグメントについて式（9）と式（10）の両方が成立する場合のみ、DATAセグメント紛失数に1だけ加算）を行うため、本例では、重複ACKフラグ（dupackと表記する）を導入している。

【0057】

そして、式（12）に示すように重複ACKフラグ（dupack）が0である場合のみ、式（13）及び式（14）の両方が成立するか否か判断する。但し

、式(13)及び式(14)は、前の式(9)及び式(10)と同じである。

$$\text{dupack} = 0 \quad \dots \text{式(12)}$$

$$\text{ACK} = \text{ack_next} \quad \dots \text{式(13)}$$

$$\text{WIN} = \text{win} \quad \dots \text{式(14)}$$

【0058】

更に、前式(12)、前式(13)及び前式(14)が全て成立する場合のみ、式(15)により発側DATAセグメント紛失数 ini_recv_drop_num に1を加算し、且つ、式(16)のように重複ACKフラグ(dupack)を1に変更するようにしている。但し、式(15)は前の式(11)と同じである。

$$\begin{aligned} \text{ini_recv_drop_num} &= \text{ini_recv_drop_num} \\ &\quad + 1 \quad \dots \text{式(15)} \end{aligned}$$

$$\text{dupack} = 1 \quad \dots \text{式(16)}$$

【0059】

ここで、重複ACKフラグ(dupack)の値は、解析モジュール3が下記のように管理する。

- (1) 重複ACKフラグの初期値を0とする。
- (2) 初期状態($\text{dupack} = 0$)又は $\text{dupack} = 0$ にて、前述の式(13)と式(14)の両方が成立するACKセグメントを始めて検出した時点で、重複ACKフラグの値を0から1に変更する($\text{dupack} = 1$)。
- (3) $\text{dupack} = 1$ の状態、重複ACKフラグの値を0から1に変更した時のACKセグメントと応答確認番号の値あるいはウィンドウサイズの値が異なるACKセグメントを検出した時点で、重複ACKフラグの値を1から0に戻す($\text{dupack} = 0$)。
- (4) $\text{dupack} = 1$ の状態では、重複ACKフラグの値を0から1に変更した時のACKセグメントと同じ値の応答確認番号とウィンドウサイズの組を持つACKセグメントを次に検出しても、重複ACKフラグの値は $\text{dupack} = 1$ のままで変更しない。

【0060】

以上のように、SYNセグメントを検出し、更に、検出したSYNセグメントと同じコネクションに属するACKセグメントを検出した場合、新たに検出したACKセグメントの応答確認番号ACK及びウインドウサイズWINの両方が次応答確認番号ack__next及び現ウインドウサイズwinに等しいか否かを判定し、応答確認番号ACK及びウインドウサイズWINの組が同じ値の2つ以上のACKセグメントについて前述の式(9)と式(10)の判定が共に肯定である毎に、今までの発側DATAセグメント紛失数に1を加算して新たな発側DATAセグメント紛失数ini__recv__drop__numを求める。

【0061】

図7に示すシーケンスにおいて、1番目のACKセグメントの応答確認番号はACK=1460、WIN=8192である。従って、最初は、次応答確認番号はack__next=1460、現ウインドウサイズはwin=8192である。重複ACKフラグ(dupack)は初期値0のままである。

【0062】

2番目のACKセグメントの応答確認番号はACK=2920、そのウインドウサイズはWIN=8192であることから、ACK≠ack__nextである(応答確認番号ACK=2920が次応答確認番号ack__next=1460に等しくない)ため、次応答確認番号はack__next=2920となり、重複ACKフラグ(dupack)は0のままである。

【0063】

3番目のACKセグメントの応答確認番号とウインドウサイズの組(ACK=2920, WIN=8192)は2番目のACKセグメントと同じ値であり、それぞれ次応答確認番号ack__next(=2920)と現ウインドウサイズwin(=8192)に一致する。従って、3番目のACKセグメントを検出した時点で、発側DATAセグメント紛失数ini__recv__drop__numは、今までの0に1が加算されて、1となる。また、重複ACKフラグは0から1へ変わる。

【0064】

4番目のACKセグメントの応答確認番号とウインドウサイズの組(ACK=

2920, WIN=8192)も2番目のACKセグメントと同じ値であり、それぞれ次応答確認番号ack__nxt (=2920)と現ウィンドウサイズwin (=8192)に一致する。しかし、この時点までは重複ACKフラグは1であるから、発側DATAセグメント紛失数の加算はない。また、重複ACKフラグは1のままである。

【0065】

5番目のACKセグメントの応答確認番号はACK=4380であり、そのウィンドウサイズはWIN=8192であり、2～4番目のACKセグメント(ACK=2920, WIN=8192)とは応答確認番号とウィンドウサイズの組、特に応答確認番号が異なる。

【0066】

一方、4番目のACKセグメントを検出した時点では、次応答確認番号はack__nxt=2920であり、現ウィンドウサイズはwin=8192である。

【0067】

従って、5番目のACKセグメントを検出した時点で、ACK≠ack__nxtである(応答確認番号ACK=4380が次応答確認番号ack__nxt=2920に等しくない)から、次応答確認番号はack__nxt=4380となり、重複ACKフラグ(dupack)は0に戻る。

【0068】

6番目のACKセグメントの応答確認番号とウィンドウサイズの組(ACK=4380, WIN=8192)は5番目のACKセグメントと同じ値であり、それぞれ次応答確認番号ack__nxt (=4380)と現ウィンドウサイズwin (=8192)に一致する。従って、6番目のACKセグメントを検出した時点で、発側DATAセグメント紛失数ini__recv__drop__numは、今までの1に1が加算されて、2となる。また、重複ACKフラグは0から1へ変わる。

【0069】

7番目と8番目のACKセグメントを検出しても、発側DATAセグメント紛失数の加算はない。即ち、

(1) 7番目のACKセグメントの応答確認番号は $ACK=5840$ であり、そのウィンドウサイズは $WIN=8192$ であり、6番目のACKセグメント($ACK=4380$, $WIN=8192$)とは応答確認番号とウィンドウサイズの組、特に応答確認番号が異なる。一方、6番目のACKセグメントを検出した時点では、次応答確認番号は $ack_next=4380$ であり、現ウィンドウサイズは $win=8192$ である。従って、7番目のACKセグメントを検出した時点で、 $ACK \neq ack_next$ である(応答確認番号 $ACK=5840$ が次応答確認番号 $ack_next=4380$ に等しくない)から、次応答確認番号は $ack_next=5840$ となり、重複ACKフラグ($dupack$)は0に戻る。

(2) 8番目のACKセグメントの応答確認番号は $ACK=5840$ であり、そのウィンドウサイズは $WIN=16384$ であり、7番目のACKセグメント($ACK=5840$, $WIN=8192$)とは応答確認番号とウィンドウサイズの組、特にウィンドウサイズが異なる。一方、7番目のACKセグメントを検出した時点では、次応答確認番号は $ack_next=5840$ であり、現ウィンドウサイズは $win=8192$ である。従って、8番目のACKセグメントを検出した時点で、 $WIN \neq win$ である(ウィンドウサイズ $WIN=16384$ が現ウィンドウサイズ $win=8192$ に等しくない)から、現ウィンドウサイズは $win=16384$ となり、重複ACKフラグ($dupack$)は0のままである。

【0070】

[着側DATAセグメント紛失数の収集]

着側DATAセグメント紛失数の収集には、モニタ可能な方向のトラヒックで、先ず、 $SYN+ACK$ セグメントを検出する(図示省略)。更に、この $SYN+ACK$ セグメントと同じコネクションに属するACKセグメントを検出する(図7参照)。

【0071】

モニタ可能な方向で、 $SYN+ACK$ セグメントに続いてACKセグメントが新たに検出された場合、この新たに検出したACKセグメントの応答確認番号 ACK 及びウィンドウサイズ WIN を、前述の別途管理している次応答確認番号 ack_next 及び現ウィンドウサイズ win とそれぞれ比較する。

【0072】

そして、先に説明した発側DATAセグメント紛失数の収集と同様、重複ACKフラグ (dupack) を導入し、式 (17)、式 (18) 及び式 (19) の全てが成立するか否か判断する。

$$\text{dupack} = 0 \quad \dots \text{式 (17)}$$

$$\text{ACK} = \text{ack_next} \quad \dots \text{式 (18)}$$

$$\text{WIN} = \text{win} \quad \dots \text{式 (19)}$$

【0073】

式 (17) に示したように重複ACKフラグが0の場合に、前記の式 (18) と式 (19) の両方が成立すれば、式 (20) に示すように着側DATAセグメント紛失数 rsp_recv_drop_num に1を加算する。つまり、応答確認番号ACKとウィンドウサイズWINの組が同じ値のACKセグメントについて2回以上DATAセグメント紛失があったと判断した場合のみ、式 (20) に示すように、今までの着側DATAセグメント紛失数に1を加算して、新たな着側DATAセグメント紛失数 rsp_recv_drop_num とする。また、式 (21) に示すように重複ACKフラグを1に変更する。

$$\begin{aligned} \text{rsp_recv_drop_num} &= \text{rsp_recv_drop_num} \\ &\quad + 1 \quad \dots \text{式 (20)} \end{aligned}$$

$$\text{dupack} = 1 \quad \dots \text{式 (21)}$$

【0074】

言い換えれば、SYN+ACKセグメントを検出し、更に、検出したSYN+ACKセグメントと同じコネクションに属するACKセグメントを検出した場合、新たに検出したACKセグメントの応答確認番号ACK及びウィンドウサイズWINの両方が次応答確認番号 ack_next 及び現ウィンドウサイズ win に等しいか否かを判定し、応答確認番号ACK及びウィンドウサイズWINの組が同じ値の2つ以上のACKセグメントについて前述の式 (18) と式 (19) の判定が共に肯定である毎に、今までの着側DATAセグメント紛失数に1を加算して新たな着側DATAセグメント紛失数 rsp_recv_drop_num を求める。

【 0 0 7 5 】

次に、図 8 (a) 、 (b) を参照して、HTTP の応答時間及びスループットの収集を説明する。

【 0 0 7 6 】

[HTTP 通信の説明]

HTTP の通信は、クライアントからの要求を表す HTTP リクエストと、それに対応するサーバからの応答を表す HTTP レスポンスにより構成される。HTTP リクエスト及び HTTP レスポンスはそれぞれ複数の DATA セグメントで構成され、通常は、テキストや画像等のコンテンツを含む HTTP レスポンスの方が、DATA セグメントの総バイト数が大きくなる。また、クライアント及びサーバからの各 DATA セグメント群に対応して、複数の ACK セグメントがサーバ及びクライアントから送られる。

【 0 0 7 7 】

[HTTP の発側応答時間及び発側スループットの収集]

図 8 (a) は HTTP の発側応答時間及び発側スループットの収集を説明するためのシーケンス図であり、これらの収集には、モニタ可能な方向のトラヒックで、SYN セグメントを検出し、更に、この SYN セグメントと同じコネクションに属する連続複数の DATA セグメント (HTTP リクエスト) 及びこの HTTP リクエストに続く連続複数の ACK セグメント (HTTP レスポンス) を検出する。

【 0 0 7 8 】

HTTP 発側応答時間 `ini_http_rsp_time` は、HTTP リクエストの最後のセグメントと HTTP レスポンスの最初のセグメントとの時間差を計算することにより、求まる。HTTP リクエストと HTTP レスポンスとの区切りは、連続して転送される DATA セグメントと ACK セグメントとの切り替わりによって検出できる。

【 0 0 7 9 】

従って、図 8 (a) に示すようにモニタ可能な方向で SYN セグメントが検出された場合、最後の DATA セグメントの検出から最初の ACK セグメントの検

出までの時刻差を、HTTP 発側応答時間 `ini_http_rsp_time` として計算する。

【0080】

一方、HTTP 発側スループット `ini_http_tpt` は、HTTP レスポンスの総バイト数と、HTTP レスポンスの最初のセグメントから最後のセグメントまでの時刻差との比とする。これは、通常 HTTP リクエストよりも HTTP レスポンスの方が DATA セグメントの総バイト数が多いので、精度の点で、HTTP レスポンスを転送している間のスループットを求めることが好ましいからである。

【0081】

そこで、図 8 (a) に示すようにモニタ可能な方向で SYN セグメントが検出された場合、HTTP レスポンスの総バイト数 `http_rsp_byt` を、最初に検出した ACK セグメントと最後に検出した ACK セグメントとの応答確認番号の差を計算して求める。また、HTTP レスポンスの最初のセグメントから最後のセグメントまでの時刻差 `http_rsp_dtime` を、最初の ACK セグメントの検出から最後の ACK セグメントの検出までの時刻差を計算して求める。

【0082】

更に、式 (22) に示すように、総バイト数 `http_rsp_byt` を時刻差 `http_rsp_dtime` で割算することにより、HTTP 発側スループット `ini_http_tpt` を求める。

$$\text{ini_http_tpt} = \text{http_rsp_byt} / \text{http_rsp_dtime} \quad \cdots \text{式 (22)}$$

【0083】

但し、HTTP 発側スループット `ini_http_tpt` の計算において、異常値を採らないように、連続して検出された ACK セグメントの数が所定の閾値 (10 程度の定数) 以上の場合のみ、スループットとして算出している。

【0084】

HTTP 発側応答時間と HTTP 発側スループットのうち、一方のみを計算す

るようにしても良い。また、HTTPリクエストの総バイト数と、HTTPリクエストの最初のセグメントから最後のセグメントまでの時刻差との比を計算して、HTTP発側スループットを求めることも可能である。

【0085】

[HTTPの着側応答時間及び着側スループットの収集]

図8(b)はHTTPの着側応答時間及び着側スループットの収集を説明するためのシーケンス図であり、これらの収集には、モニタ可能な方向のトラヒックで、SYN+ACKセグメントを検出し、更に、このSYN+ACKセグメントと同じコネクションに属する連続複数のACKセグメント(HTTPリクエスト)及びこのHTTPリクエストに続く連続複数のDATAセグメント(HTTPレスポンス)を検出する。

【0086】

HTTP着側応答時間 $rsp_http_rsp_time$ も、HTTPリクエストの最後のセグメントとHTTPレスポンスの最初のセグメントとの時間差を計算することにより求まる。HTTPリクエストとHTTPレスポンスとの区切りは、連続して転送されるACKセグメントとDATAセグメントとの切り替わりによって検出できる。

【0087】

そこで、図8(b)に示すようにモニタ可能な方向でSYN+ACKセグメントが検出された場合、最後のACKセグメントの検出から最初のDATAセグメントの検出までの時刻差を、HTTP着側応答時間 $rsp_http_rsp_time$ として計算する。

【0088】

HTTP着側スループット rsp_http_tp は、HTTPレスポンスの総バイト数と、HTTPレスポンスの最初のセグメントから最後のセグメントまでの時刻差との比とする。これも、通常HTTPリクエストよりもHTTPレスポンスの方がDATAセグメントの総バイト数が多いので、精度の点で、HTTPレスポンスを転送している間のスループットを求めることが好ましいからである。

【0089】

そこで、図8(b)に示すようにモニタ可能な方向でSYN+ACKセグメントが検出された場合、HTTPレスポンスの総バイト数 $http_rsp_byt$ を、最初に検出したDATAセグメントのシーケンス番号と最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算して求める。また、HTTPレスポンスの最初のセグメントから最後のセグメントまでの時刻差 $http_rsp_dtime$ を、最初のDATAセグメントの検出から最後のDATAセグメントの検出までの時刻差を計算して求める。

【0090】

更に、式(23)に示すように、総バイト数 $http_rsp_byt$ を時刻差 $http_rsp_dtime$ で割算することにより、HTTP着側スループット rsp_http_tpt を求める。

$$rsp_http_tpt = http_rsp_byt / http_rsp_dtime \quad \dots \text{式(23)}$$

【0091】

HTTP着側スループット rsp_http_tpt の計算において、異常値を採らないように、連続して検出されたACKセグメントの数が所定の閾値(10程度の定数)以上の場合のみ、スループットとして算出している。

【0092】

この場合も、HTTP着側応答時間とHTTP着側スループットのうち、一方のみを計算するようにしても良い。また、HTTPリクエストの総バイト数と、HTTPリクエストの最初のセグメントから最後のセグメントまでの時刻差との比を計算して、HTTP着側スループットを求めることも可能である。

【0093】

次に、図9(a)、(b)を参照して、FTPのスループットの収集を説明する。

【0094】

[FTP通信の説明]

FTPの通信は、制御データを転送するための制御用TCPコネクションと、ファイルを転送するためのデータ用TCPコネクションを用いて行われる。このうち、データ用TCPコネクションでは、サーバからクライアントへ大量のバイト数（大量のDATAセグメント）が転送される可能性があるため、スループットの計算を行うことが可能である。サーバからの大量のDATAセグメントに対応して、クライアントからサーバへACKセグメントが大量に送られる。但し、前述のように、データ用TCPコネクションではサーバがSYNセグメントを送信し、クライアントがSYN+ACKセグメントを送信する。従って、FTPのスループットを収集する場合、発側とはSYN+ACKセグメントの送信側となり、着側とはSYNセグメントの送信側となる。

【0095】

[FTPの発側スループットの収集]

図9（a）はFTPの発側スループットの収集を説明するためのシーケンス図であり、これの収集には、モニタ可能な方向のトラヒックで、SYN+ACKセグメントを検出し、更に、このSYN+ACKセグメントと同じコネクションに属する連続複数のACKセグメントを検出する。

【0096】

FTP発側スループット ini_ftp_tp は、ファイルの総バイト数と、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差との比とする。

【0097】

そこで、図9（a）に示すようにモニタ可能な方向でSYN+ACKセグメントが検出された場合、ファイルの総バイト数 ftp_byt を、最初に検出したACKセグメントと最後に検出したACKセグメントとの応答確認番号の差を計算して求める。また、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差 ftp_time を、最初のACKセグメントの検出から最後のACKセグメントの検出までの時刻差を計算して求める。

【0098】

更に、式（24）に示すように、総バイト数 ftp_byt を時刻差 $ftp_$

d t i m eで割算することにより、FTP発側スループット $i n i_f t p_t p t$ を求める。

$$i n i_f t p_t p t = f t p_b y t / f t p_d t i m e$$

…式(24)

【0099】

但し、FTP発側スループット $i n i_f t p_t p t$ の計算において、異常値を採らないように、連続して検出されたACKセグメントの数が所定の閾値(10程度の定数)以上の場合のみ、スループットとして算出している。

【0100】

[FTPの着側スループットの収集]

図9(b)はFTPの着側スループットの収集を説明するためのシーケンス図であり、これの収集には、モニタ可能な方向のトラヒックで、SYNセグメントを検出し、更に、このSYNセグメントと同じコネクションに属する連続複数のDATAセグメントを検出する。

【0101】

FTP着側スループット $r s p_f t p_t p t$ も、ファイルの総バイト数と、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差との比とする。

【0102】

そこで、図9(b)に示すようにモニタ可能な方向でSYNセグメントが検出された場合、ファイルの総バイト数 $f t p_b y t$ を、最初に検出したDATAセグメントのシーケンス番号と最後に検出したDATAセグメントのシーケンス番号にそのユーザデータ長を加算した値との差を計算して求める。また、ファイル転送の最初のセグメントから最後のセグメントまでの時刻差 $f t p_d t i m e$ を、最初のDATAセグメントの検出から最後のDATAセグメントの検出までの時刻差を計算して求める。

【0103】

更に、式(25)に示すように、総バイト数 $f t p_b y t$ を時刻差 $f t p_d t i m e$ で割算することにより、FTP着側スループット $r s p_f t p_t p t$

p t を求める。

$$r s p_f t p_t p t = f t p_b y t / f t p_d t i m e$$

…式 (25)

【0104】

但し、FTP着側スループット $r s p_f t p_t p t$ の計算においても、異常値を採らないように、連続して検出されたDATAセグメントの数が所定の閾値（10程度の定数）以上の場合のみ、スループットとして算出している。

【0105】

以上のように、各種の情報収集において、SYNセグメントを検出したかSYN+ACKセグメントを検出したかによって発側のトラフィック統計情報か着側のトラフィック統計情報かを区別することにより、つまり、モニタした方向がSYNセグメントの送信側かSYN+ACKセグメントの送信側かを区別することにより、クライアントからの通信かサーバからの通信かを識別することができる。

【0106】

【発明の効果】

以上説明したように、本発明によれば、片方向のトラフィックしか取得できない場合においても、反対方向からのトラフィックが推定可能であり、より多くのトラフィック統計情報を収集することができる。従って、本発明はインターネットのバックボーン回線のトラフィック調査に極めて有用である。

【図面の簡単な説明】

【図1】

本発明の実施形態例として、双方向トラフィック統計情報収集装置の構成例を示す図。

【図2】

本発明の実施形態例として、発側データ転送量を収集する場合のシーケンス例を示す図。

【図3】

本発明の実施形態例として、着側データ転送量を収集する場合のシーケンス例を示す図。

【図 4】

本発明の実施形態例として、発側データ受信量を収集する場合のシーケンス例を示す図。

【図 5】

本発明の実施形態例として、着側データ受信量を収集する場合のシーケンス例を示す図。

【図 6】

本発明の実施形態例として、データ受信量を収集する場合のシーケンス例、並びに、重複バイトの例を示す図。

【図 7】

本発明の実施形態例として、セグメント紛失数を収集する場合のシーケンス例を示す図。

【図 8】

本発明の実施形態例として、H T T P の応答時間及びスループットを収集する場合のシーケンス例を示す図。

【図 9】

本発明の実施形態例として、F T P のスループットを収集する場合のシーケンス例を示す図。

【符号の説明】

- 1 双方向トラヒック統計情報収集装置
- 2 セグメント取得モジュール
- 3 解析モジュール
- 4 蓄積モジュール
- 5 状態遷移表の格納部
- 6 トラヒック統計情報の格納部
- 7、8、9 端末
- 10 インターネット
- 11、12、13、14 ルータ
- 15 バックボーン

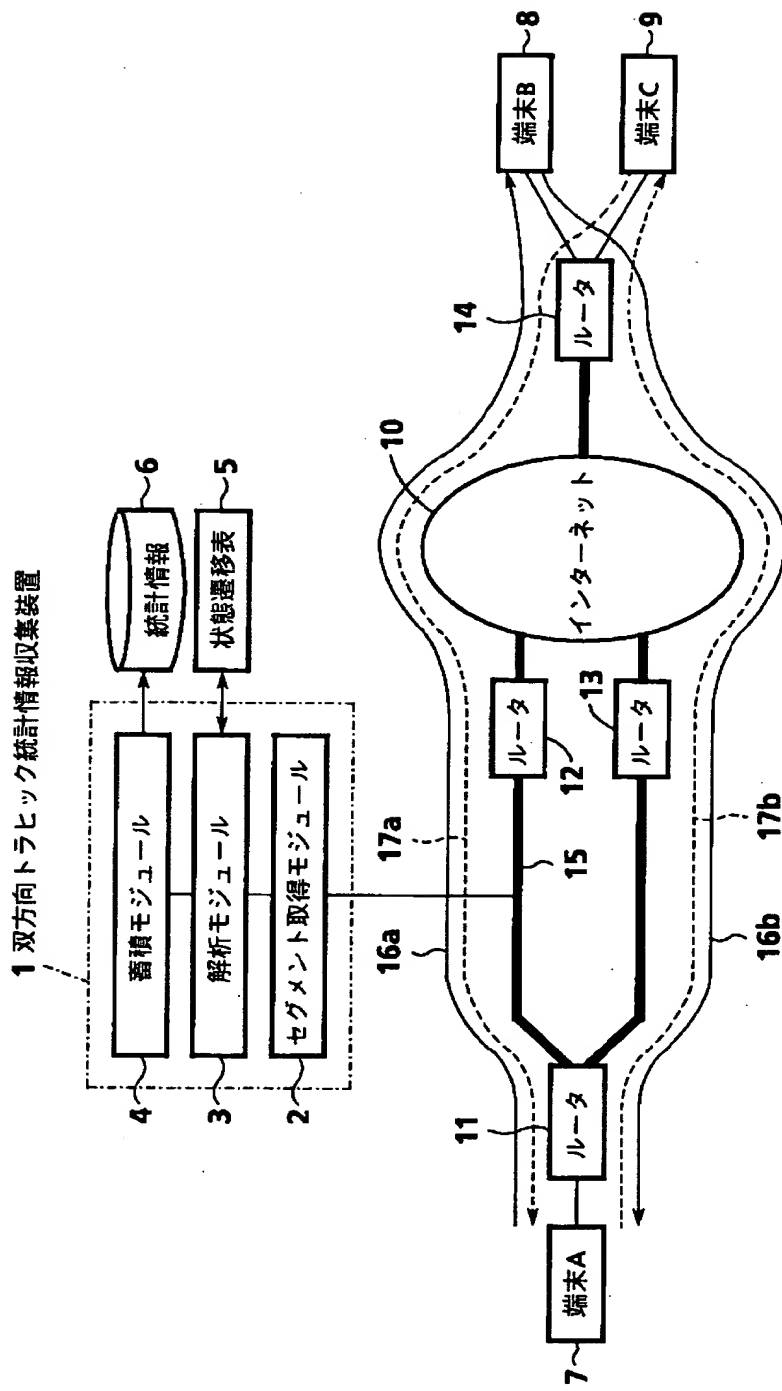
1 6 a、1 7 a セグメントの取得可能なトラヒック

1 8、1 9 重複バイト部分

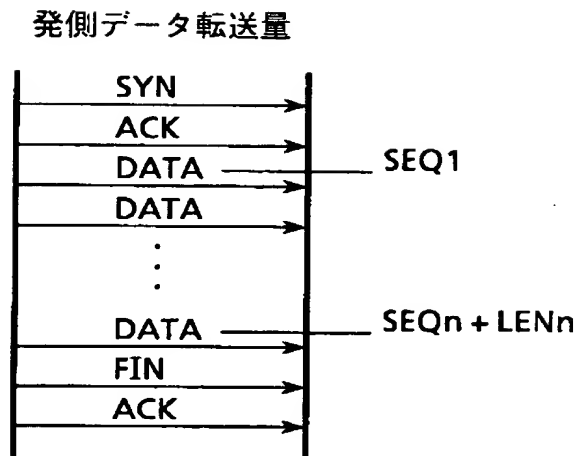
【書類名】

図面

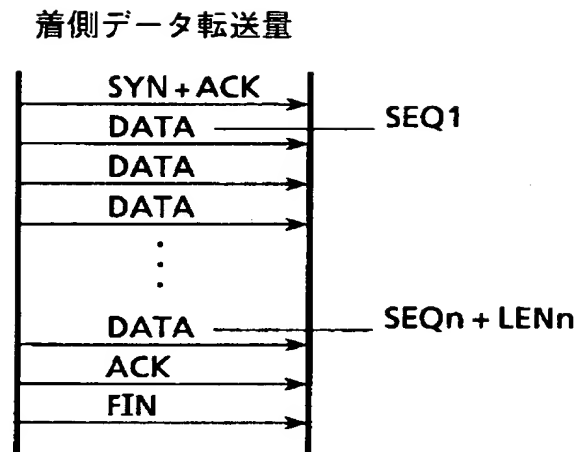
【図 1】



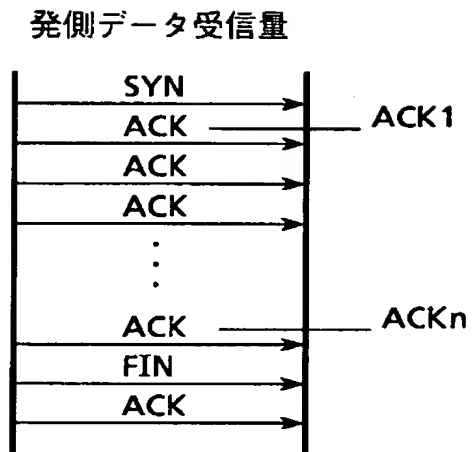
【図 2】



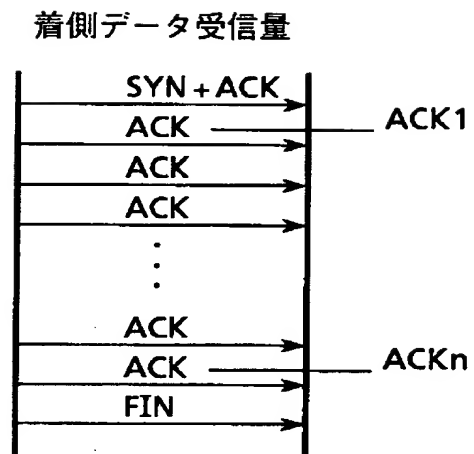
【図 3】



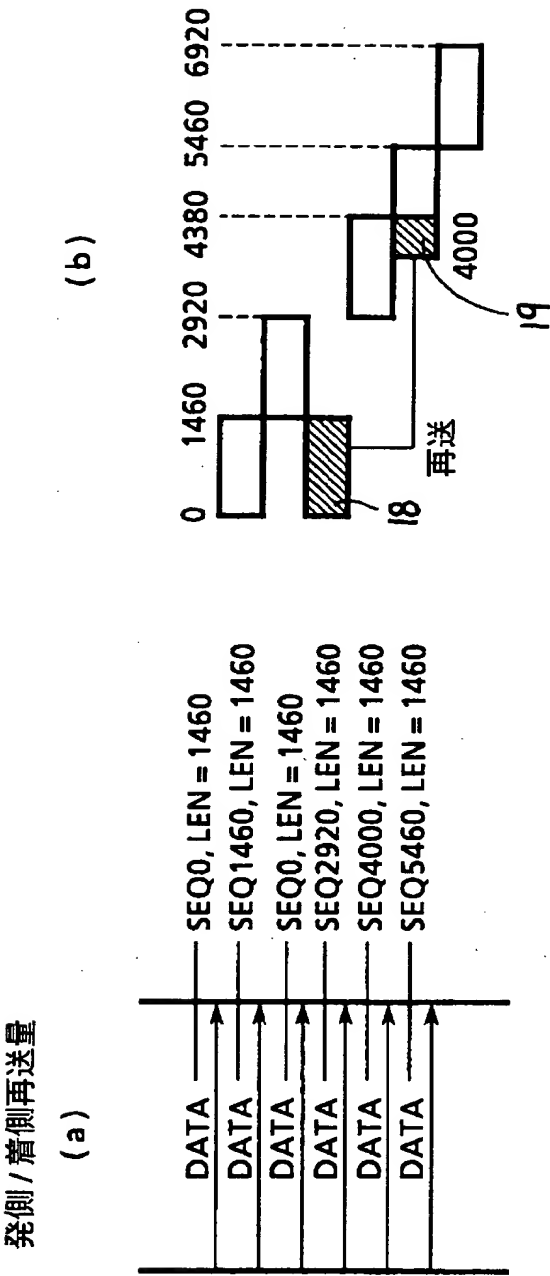
【図 4】



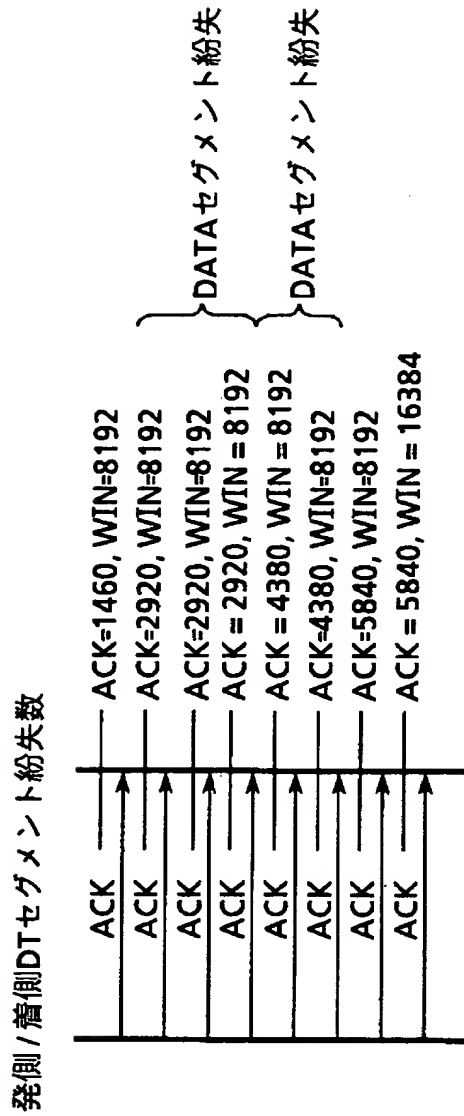
【図 5】



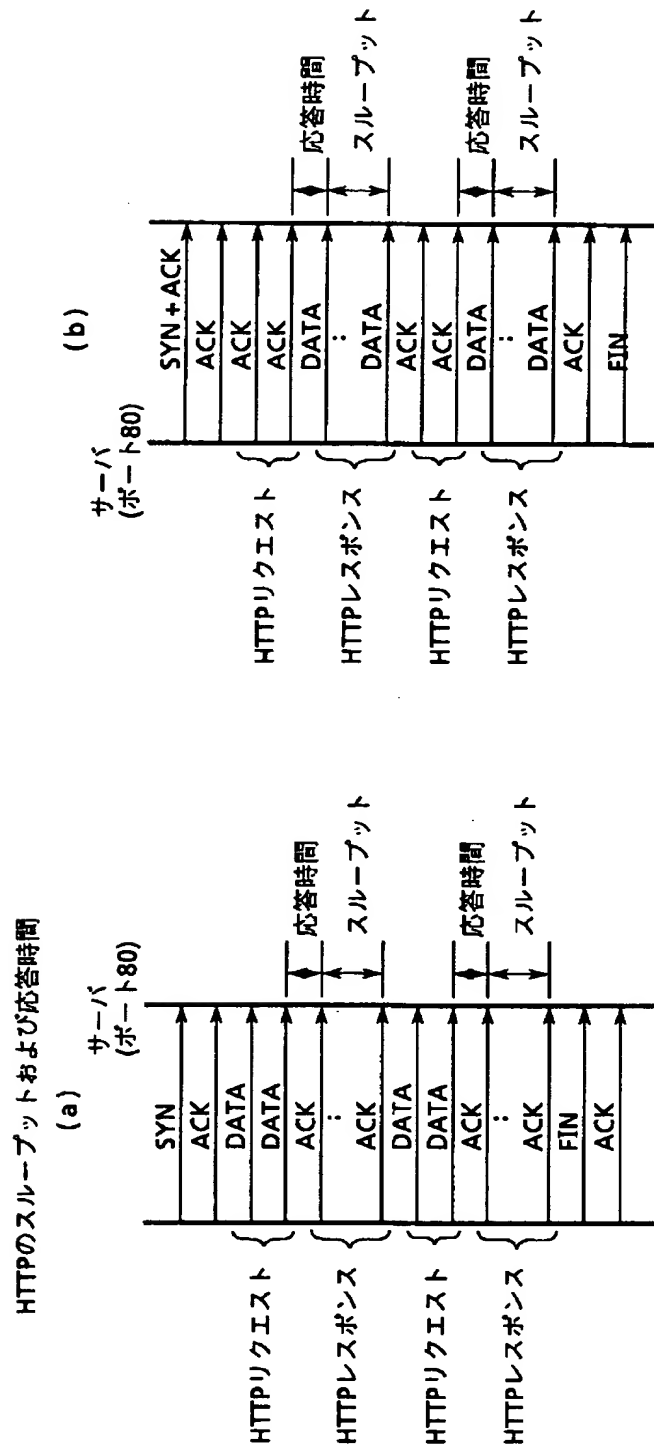
【図 6】



【図 7】

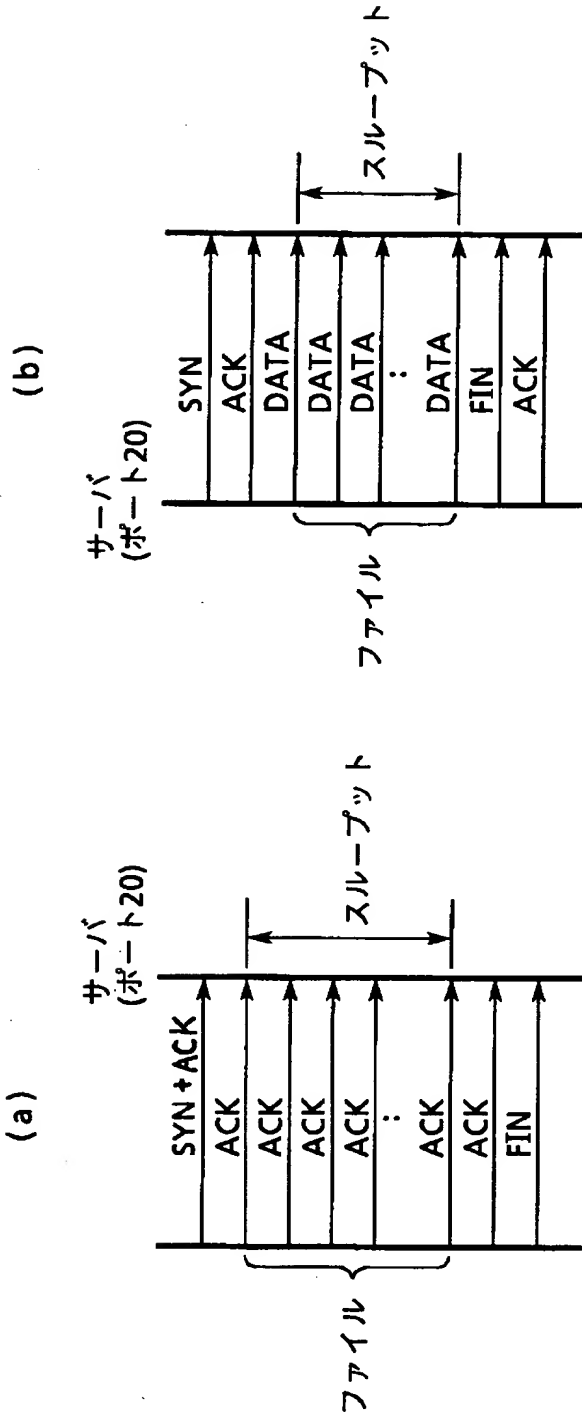


【図 8】



【図 9】

FTPのデータ用TCPコネクションのスループット



【書類名】 要約書

【要約】

【課題】 片方向のトラヒックしか取得できない場合においても、トラヒックが取得できなかった方向のトラヒック統計情報を収集できること。

【解決手段】 SYNセグメントが検出された場合、連続して検出されたDATAセグメント数を計算し、発側転送セグメント数とする。また、連続して検出されたDATAセグメントのうち、最初のDATAセグメントのシーケンス番号SEQ1と、最後のDATAセグメントのシーケンス番号SEQnと、最後のDATAセグメントのユーザデータ長LENnを用い、 $ini_sdt = (SEQn + LENn) - SEQ1$ を計算し、発側転送バイト数ini_sdtを求める。

【選択図】 図2

出 願 人 履 歴 情 報

識別番号 [000001214]

1. 変更年月日 1998年12月 3日
[変更理由] 名称変更
住 所 東京都新宿区西新宿2丁目3番2号
氏 名 ケイディディ株式会社